# On Evaluation of a Population of Bayesian Networks

Hoai-Tuong Nguyen, Gérard Ramstein, Philippe Leray

# On Evaluation of a Population of Bayesian Networks


Nguyen Hoai Tuong
Polytechnic School
of University of Nantes
Laboratory of Computer Science
of Loire-Atlantique (LINA)
Email: hoai-tuong.nguyen@univ-nantes.fr

Ramstein Gérard
Polytechnic School
of University of Nantes
Laboratory of Computer Science
of Loire-Atlantique (LINA)
Email: gerard.ramstein@univ-nantes.fr

Leray Philippe
Polytechnic School
of University of Nantes
Laboratory of Computer Science
of Loire-Atlantique (LINA)
Email: philippe.leray@univ-nantes.fr



***Abstract*—Two new evaluation of quality approaches for a population of Bayesian networks (BN) are proposed in this paper. The first approach relies on the use of statistical principle with application of well-known evaluation methods. The other bases on epsilon-quasi essential graph (QEG), an extension of essential graph (EG), that is a presentative graph for all BN of the population. In QEG, each edge is statistically weighted in two parts: (1) undirected part that represents the power of the relationship; (2) arrow part that represents the reliability of the orientation. Results of application to the both simulated and real-world problems show that these proposed approaches are the others helpful solutions for the problem of edge orientation and for the visualization of results of evaluation methods.**


## I. INTRODUCTION

Bayesian networks (BN) are one of the powerful graphical probabilistic models for representing and analyzing uncertainty knowledge. One of the strong interest of BN is the ability to learn the best structure that gives the best fit to the observational data. In the literature, the most of BN structure learning algorithm keeps only one the best structure. But with some others algorithms, the final result can be many BN instead of one, for example: *evolutionary algorithms* [1], *Bootstrap* [2]. The issue is if we keep the best structure with an evaluation criteria, the others can be also the best structures with another criteria. Thus, we need the good evaluation method that can evaluate also the others structures.

The evaluation of quality of BN is a crucial step in BN structure learning. Although, there are various proposed methods in the literature [3]–[6], but the problem of poverty of evaluation and visualization method for a population of structures still remains to be solved. To our knowledge, we can only cite here a related work presented by the research team of Imoto, Kamimura that based on the Boostrap analysis [7], [8]. Due to the cause of dimensionality and complexity of the data, this work still has the problem of the orientation among edges. They do not take into account the problem of Markov-equivalent class. Moreover, if the orientation is defined only by the edge intensity it can be cause the problem of cycle and irreversible edges. Therefore, to describe the most *common* properties of all population, the theory and the methodology must be expected to not only a statistical point of view but also some typical rules for orientation of BN structure. Our purpose is to establish two new approaches for measuring and describing more clearly the relationships among variables in order to estimate precisely the quality of the population: the first approach relies on the mean of quality of all BN. The quality of each BN can be estimated by one of these well-known evaluation methods: *score, Kullback Leibler divergence, sensibility/specificity, edit distance*. This technique allows us to identify directly a global quality of population by the available evaluation methods. The other approach bases on *epsilon-quasi essential graph* (QEG). QEG is an extension of essential graph (EG). The QEG based evaluation algorithm allows to construct statistically a *unique representative* graph for all population of BN. In QEG, each edge is weighted in two parts: (1) undirected part that represents the power of the relationship; (2) arrow part that represents the reliability of the orientation. This technique allows us to obtain firstly a representant of population. Then we can evaluate the quality of population via this representant.

## II. METHOD

### A. Basics concepts

**Definition 1**: **Directed graph** is defined by a couple $G = (V, E)$ where:

- $V$ is a definite set of vertices;
- $E \subseteq V \times V$ is a set of couples of vertices called edges.

**Definition 2: Markov condition** *is defined as following: Each variable $X_i$ is conditionally independent to a set of its non-descendants, $NonDesc(X_i)$, given its parents, $Pa(X_i)$, therefore we can note $P(Xi|Pa(X_i), NonDesc(X_i)) = P(X_i|Pa(X_i))$.*

**Definition 3**: $\mathcal{B} = (G, \theta)$ *is* ***a Bayasian network*** *if $G = (X, E)$ is a directed acyclic graph (DAG) that the vertices represent a set of random variables $X = (X_1, ..., X_n)$ and $\theta_i = [P(X_i|Pa(X_i))]$ is the matrix of conditional probabilities of vertices $i$ given its parents in $G$ and the couple $(G, \theta)$ verifies the Markov condition. The joint probability distribution on $X$ is defined by :*

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i)) \qquad (1)$$

### B. Statistical approach

The idea of this approach is to reuse evaluation methods for one BN (presented below) in order to evaluate the quality of each BN of the population and then we calculate the mean of all obtained results as the quality of a population of BN. This approach can be applied by two ways: (1) *without equivalence class:* we can evaluate directly the quality of each BN without taking into account the problem of Markov-equivalent class. By this way, we can apply the **score** based method and **Kullback-Leibler divergence** based method; (2) *with equivalence class:* we need to take into account the problem of Markov-equivalent class before evaluating the quality of each the BN. That means we have to transform all BN to essential graph (EG) (see section II.B.5). With this way, we can use **sensitivity/specificity** based method and **edit distance** based method.

*1) Score based method:* The idea of the score is simple: It is the posterior probability distribution conditioned to the available data; The best network is the one that maximizes this score. There are different kinds of scores: entropy and information based [9], the minimum description length based [10] and Bayesian approaches [11].

The score-based method is one of the fondamental methods to evaluate the quality of BN. There are various researches to improve its performance. However, it's hard to distinguish the performance between different types of score [12]. The choice of score depends on the specific context of the application. Moreover, in the case of Markov equivalence between BN, the score is incapable of distinguishing the BNs in the same equivalent class [13] (section *II.B.5*). This disadvantage becomes important in the case of comparison of the structural diffrence between two networks.

*2) Kull-back Leibler divergence based method:* Kullback-Leibler (KL) divergence is a measure of dissimilarity between two probability distributions [14]. It is used to measure the difference between the probability distribution of learned network , $P_{\mathcal{B}}$ and the golden network, $P_{\mathcal{B}_0}$. If two probability distributions are discrete, this divergence is defined by:

$$D_{KL}(\mathcal{B},\mathcal{B}_0)=\sum_{X}P_{\mathcal{B}}(X)log\frac{P_{\mathcal{B}}(X)}{P_{\mathcal{B}_0}(X)} \tag{2}$$

where $X$ is a set of random variables $X = X_1, ..., X_n$.

This formula is used with a convention that: *0 log 0 = 0*. The value of KL is nonnegative and equal to zero when the laws $P_{\mathcal{B}}$ and $P_{\mathcal{B}_0}$ are identical. For two continuous probability distributions, $P_{\mathcal{B}}$ and $P_{\mathcal{B}_0}$, this measure is defined by an integral.

KL divergence is a often used method in the probabilistic graphical models learning, including Gaussian models. However, the KL divergence suffers from its limitations when there are many variables, the computational complexity of this measure becomes large.

*3) Sensibility/specificity based method:* Inspired for statistics, the evaluation method for a BN based on sensitivity/specificity is a measure of the ability to match the edges presented learned BN to golden BN. The sensibility indicates the capacity of the learning algorithm to identify an edge presented in golden structure. The specificity indicates the capacity of the learning algorithm to identify an edge absent in golden structure. More precisely, the sensibility and specificity can be calculated as follows:

$$\textit{Sensibility}=\frac{TP}{TP+FN} \tag{3}$$

$$\textit{Specificity}=\frac{TN}{TN+FP} \tag{4}$$

where $TP$ (*true positive*) = number of edges *present* in the both learned and golden network; $TN$ (*true negative*) = number of edges *missing* in the both learned and golden network; $FP$ (*false positive*) = (number of arcs in the learned network - TP); $FN$ (*false negative*) = (number of edges missing in the golden network - TN);

This measure is often used in the literature. It is capable of distinguishing the BN in the same equivalence class. So it's interesting to study the differential structure of BN. However, this advantage also becomes a limitation. In fact, the structural difference between the BN of the same equivalence class is regarded as errors (section *II.B.5*).

*4) Edit distance based method:* If the sensitivity/specificity allows to describe the *similarity* between the learned and golden BN, the edit distance is interested in *dissimilarity* between them. It calculates the cost of modifying operations to transform the learned graph to golden graph. To calculate the cost of edit operations, you should use the concept of *"matching"* that is defined as follows [15]: Given $G(V,E)$ and $G_0(V_0,E_0)$ represent respectively the DAG of learned BN, $\mathcal{B}$ and golden BN, $\mathcal{B}_0$. The (*matching*) of $G$ to $G_0$ is defined respectively by the bijective functions as following:

- $f_v : V' \to V_0'$, where $V' \subseteq V$ is a subset of vertices of $G$ and $V_0' \subseteq V_0$ is a subset of vertices of $G_0$;
- $f : E' \to E_0'$, where $E' \subseteq E$ is a subset of edges of $G$ et $E_0' \subseteq E_0$ is a subset of edges of $G_0$;

The cost associated with each edit operation to make of $G(V,E)$ in $G_0(V_0,E_0)$ can be defined as following:

- *Addition*:
  1) the cost of adding a node is defined by:

$$A_i(X_i)=\begin{cases}1 & \text{if } X_i \in V_0 - V_0' \\ 0 & \text{otherwise}\end{cases} \tag{5}$$

  2) the cost of adding an edge is defined by:

$$A_{ij}(E_{ij}(X_i,X_j))=\begin{cases}1 & \text{if } E_{ij}(X_i,X_j) \in E_0 - E_0' \\ & \text{and } E_{ij}(X_i,X_j) \notin E \\ & \text{where } X_i, X_j \in V_0 - V_0' \\ 0 & \text{otherwise}\end{cases} \tag{6}$$

- *Deletion*:

1) the cost of deleting a node is defined by:

$$D_i(X_i) = \begin{cases} 1 & \text{si } X_i \in V - V' \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

2) the cost of deleting an edge is defined by:

$$D_{ij}(E_{ij}(X_i, X_j)) = \begin{cases} 1 & \text{if } E_{ij}(X_i, X_j) \in E - E' \\ & \text{and } E_{ji}(X_j, X_i) \notin E_0 \\ & \text{where } X_i, X_j \in V - V' \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

- *Inversion*: the cost of reversing an edge is defined by:

$$I_{ij}(E_{ij}(X_i, X_j)) = \begin{cases} 1 & \text{if } E_{ij}(X_i, X_j) \in E_0 - E_0' \\ & \text{and } E_{ji}(X_j, X_i) \in E \\ & \text{where } X_i, X_j \in V_0 - V_0' \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The edit distance is the total of all edit costs above [15]. If the total cost of edit operations calculated only on the edges is called the *Structural Hamming Distance* SHD) [6].

This is a well-known evaluation method that easy to calculate. However, as the approach based on sensitivity/specificity, the equivalence of certain differently oriented edges in the sense of Markov is also considered as errors (section *II.B.5*).

*5) Problem of the Markov-equivalent class: Firstly*, as previously indicated, when two BN encodes the same conditional independence, the score is impossible to distinguish them (because they have the same score) (section *II.B.1*). These networks are equivalent under Markov condition (Definition 2). This is the problem of the Markov-equivalent class [13]. For example, given three variables $X, Y, Z$. Suppose that we have four BN $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$ and their conditional probabilities are presented respectively as follows:

($\mathcal{B}_1$) X→Y→Z $\quad P_{\mathcal{B}_1}(X,Y,Z) = P(X) * P(Y|X) * P(Z|Y)$
($\mathcal{B}_2$) X←Y→Z $\quad P_{\mathcal{B}_2}(X,Y,Z) = P(X|Y) * P(Y) * P(Z|Y)$
($\mathcal{B}_3$) X←Y←Z $\quad P_{\mathcal{B}_3}(X,Y,Z) = P(X|Y) * P(Y|Z) * P(Z)$
($\mathcal{B}_4$) X→Y←Z $\quad P_{\mathcal{B}_4}(X,Y,Z) = P(X) * P(Y|X,Y) * P(Y)$

According to the definition of conditional probability, we have: $P(X|Y) * P(Y) = P(Y|X) * P(X)$ et $P(Z|Y) * P(Y) = P(Y|Z) * P(Z)$. So, $P_{\mathcal{B}_1}(X,Y,Z) = P(X) * P(Y|X) * P(Z|Y) = P(X|Y) * P(Y) * P(Z|Y) = P_{\mathcal{B}_2}(X,Y,Z)$ and $P_{\mathcal{B}_2}(X,Y,Z) = P(X|Y)*P(Y)*P(Z|Y) = P(X|Y) * P(Y|Z) * P(Z) = P_{\mathcal{B}_3}(X,Y,Z)$. So, $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ are equivalent in the sense of Markov (see Definition 2). But, with $P_{\mathcal{B}_4}(X,Y,Z)$, $P(Y|X,Y)$ can not be simplified. So, $\mathcal{B}_4$ is not equivalent with respect to the three others.

*Secondly*, The methods based on *sensitivity/specificity* or *edit distance* consider the difference between equivalent BN as errors or the costs of edit operations. Indeed, in the example above, there are some differences in orientation of edges in the three RB equivalents $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$. Instead of taking into account as equivalent arcs, these methods count as a *false* (*positive* or *negative*) or cost of reversal operations. This is also anther type of problem associated to the Markov-equivalent class.

Before describing the solution for the problem of the Markov-equivalent class, we give some useful definitions and theorems:

**Definition 4:** Two DAGs are ***Markov-equivalent*** if they have the same set of Markov conditions.

**Definition 5:** ***Skeleton*** is the undirected graph obtained by removing directions from all arcs.

**Definition 6:** ***V-structure*** is one of a kind of Markov constraints for three variables $X, Y, Z$ where: (1) $X \to Y$ and $Z \to Y$; (2) $X$ and $Z$ are not adjacent.

**Theorem 1** [16] Two DAGs are equivalent if and only if they have the same skeleton and the same v-structures.

**Definition 7:** ***(Compelled edge)*** A directed edge $x \to y$ is a *compelled* edge in $G$ if for every DAG $G'$ equivalent to G, $x \to y$ exists in $G'$.

A consequence of Theorem 1 is that any edge participating in a v-structure is *compelled*. However every compelled edge not necessarily participates in a v-structure. This is the case of *inferred edge*.

**Definition 8:** ***Inferred edge*** is a directed and irreversible edge because its edge inversion create another v-structure.

A solution initially proposed by Chickering [13] solves this problem through a graph *unique* representing all equivalent BN. This graph is called ***essential graph***.

**Definition 9:** ***Essential graph***[1] (EG) is a graph that retains all the directed edges that are common to all the Markov equivalent graphs and removes the direction of the remaining edges.

In order to solve the problem of the Markov-equivalent class, we transform firstly the DAG to EG [13]. Then, with the obtained EG, we can apply normally all evaluation methods presented above (except the edit distance based method, we must modified some edit costs to calculate the distance between two EGs [6]). The quality of the essential graph is also the quality of its original BN.

### C. Epsilon-quasi Essential graph based approach

The idea of epsilon-quasi essential graph (QEG) based approach, that we refer to as ***EG-EVAL***, is to take into account firstly the problem of the Markov equivalence class and then we summarize all obtained essential graphs of the population into a *unique presentative* essential graph, called *epsilon-quasi essential graph* (QEG). In QEG, each edge is weighted in two parts: (1) undirected part that represents the power of the relationship defined by its occurrence frequency; (2) arrow part that represents the reliability of the orientation defined by the mean of occurrence frequencies of v-structures that contain it. The EG-EVAL algorithm consists of two stages: *construct EG of population* and *evaluate EG*:

In the first stage, we choose any DAG from the population and transform it to EG. Several researchers (Verma and Pearl [16], Meek [18] and Andersson et al. [17]), present *rule-based* algorithms to implement DAG-to-EG. The idea is as follows:

[1] *Note that, in the literature, essential graph is also called as "completed PDAG" (CPDAG) [17] or "maximally oriented graphs" [18].*

1) construct *skeleton* by removing the direction of every edge in a DAG
2) direct and add *v-structures* to obtained skeleton
3) direct and add *inferred edges* to obtained skeleton (after adding V-structures) by applying a set of rules that transform undirected edges into directed edges. In this work, we use three Meek rules proofed in [18]:

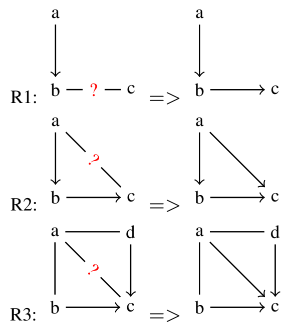



**FIGURE 1:** Three fondamental Meek rules in an essential graph [18]: (R1) If the edge were oriented in the opposite direction there would be a new *v-structure*; (R2) If the edge were oriented in the opposite direction there would be a new *cycle*; (R3) If the edge were oriented in the opposite direction then by two applications of rule R2 on $triple(a, b, c)$ and $triple(a, d, c)$ there would be two new *v-structures*;

In the second stage, we evaluate the quality of this EG by methods presented in VI such as: *sensibility/specificity, edit distance*. This is also the quality of the population.

The next pseudo-code represents the EG-based algorithm for resuming of a population of *equivalent* BNs, **EG-EVAL**:

---

**Algorithme 5.** EG-EVAL(eDAG_POP)

---

**Require:** Population of equivalent DAGs, $eDAG\_POP$.
**Ensure:** Quality of $eDAG\_POP$.

$//$**STEP 1**: $DAG => EG\_POP$

```
EG_POP← ∅;
Choose randomly a DAG, 𝒢, from eDAG_POP
//transform DAG to essential graph
EG_POP← DAG2EG(𝒢);
```

$//$**STEP 2**: Evaluation of $EG\_POP$

```
return  Quality of eDAG_POP ← EVAL(EG_POP)
//EVAL can be one of methods presented in VI.
```

---

*1) Epsilon-Quasi essential graph based (QEG-EVAL) approach for a population of random BN :* In this section, we describe an evaluation algorithm based on EG-based-EVAL for a *random* population of BNs structures, that we refer to as ***QEG-EVAL***. The QEG-EVAL takes as input a population of random DAGs, and outputs the quality of this population via an *epsilon-quasi essential graph* (QEG) representation of population.

**Definition 10:** The **epsilon-quasi essential graph** (QEG) is an essential graph that represents statistically *almost* common properties of a population of random DAGs. All small frequency edges are eliminated by a threshold $\epsilon > 0.5$.

Each undirected edge of QEG is labeled by a weight that's its occurrence frequency in the population. Each directed edge of QEG is weighted in two parts: (1) a weight represents the occurrence frequency of *undirected part*; (2) another weight represents the occurrence frequency of *arrow part* associated to the mean weight of the V-structures that contain it (see section $II.E$ for details).

Inspired from EG-EVAL, the QEG-EVAL consists also of two stages: *construct QEG of population* and *evaluate QEG*. In the first stage, there are three steps:

1) construct *skeleton* for population by adding one *weight-undirected* edge for each direct dependencies found in population of DAG.

   The undirected-edge weight is its occurrence frequency in the population. In order to control noisy-edges with the small frequency, we apply a threshold $\epsilon > 0.5$. This threshold is superior to 0.5 to avoid the confliction of direction that issues in the next step;

2) direct and add *v-structures* to obtained skeleton by each *triple* of edges formed a V-structure.

   The weight of each V-structure $a \rightarrow b \leftarrow c$ is the total by three parts: the occurrence frequency of $a \rightarrow b \leftarrow c$, one half of the occurrence frequency of $a \rightarrow b$ and one half of the occurrence frequency of $b \leftarrow c$. By this way, the weight of v-structure is systematically smaller than the weight of its pair-edges. So, to avoid to eliminate the small frequency v-structures, their weight must be normalized by divide this total for $Max(w(a - b), w(b - c))$; In order to control noisy-v-structures with the small weight, we apply a threshold $\epsilon > 0.5$. This threshold is superior to 0.5 to avoid the confliction of direction when two v-structures are created from the same edge but oriented differently;

3) direct and add *inferred edges* to obtained skeleton (after adding V-structures) by Meek rules [18]. The weight of each inferred edge is calculated as following:

   - For Meek rule $R1$: If $w(a \rightarrow b) = 1.0$ and $w(b - c) = 1.0$ then $w(b \rightarrow c) = 1.0$. If $w(a \rightarrow b) * w(b - c) > \beta$ then $w(b \rightarrow c) = w(a \rightarrow b)$
   - For Meek rule $R2$: If $w(a \rightarrow b) * w(b \rightarrow c) * w(a - c) = 1.0$ then $w(a \rightarrow c) = 1.0$. If $w(a \rightarrow b) * w(b \rightarrow c) * w(a - c) > \beta$ then $w(a \rightarrow c) = w(a \rightarrow b) * w(b \rightarrow c)$
   - For Meek rule $R3$: If $w(b \rightarrow c) * w(d \rightarrow c) * w(a - b) * w(a - d) * w(a - c) = 1.0$ then $w(a \rightarrow c) = 1.0$. If $w(b \rightarrow c) * w(d \rightarrow c) * w(a - b) * w(a - d) * w(a - c) > \beta$ then $w(a \rightarrow c) = w(b \rightarrow c) * w(b \rightarrow c)$

In the second stage, we evaluate the quality of this QEG by extended version of edit distance based evaluation methods for QEG presented nextly in $II.D$. This is also the quality of the population. The next pseudo-code represents the evaluation algorithm for a population of random BNs, **QEG-EVAL**:

**Algorithme 6.** QEG-EVAL($rndDAGPOP, \epsilon$)

**Require:** A population of random DAGs and threshold, $\epsilon$, to filter edges and v-structures with the small weight ($\alpha > 0.5$).
**Ensure:** Quality of $rndDAG\_POP$.

$N \leftarrow |rndDAGPOP|$;
*//transform all DAGs to essential graphs*
**for** $i = 1$ to $N$ **do**
  $EG_i \leftarrow DAG2EG(\mathcal{G}_i)$;
**end for**
*//Union all skeletons of EGs into one weight-skeleton UG*
$[UG, w_1(u.edges)] \leftarrow Union\{Skeleton(EG_i)\}$;
*//filter noisy-undirected-edges of UG and create QEG_POP*
$QEG\_POP \leftarrow \{u.edges \in UG / w_1(u.edges) > \epsilon\}$;
*//add and set weight v-structure for QEG_POP*
**for** $\forall triple(a, b, c)/a - b - c \in QEG\_POP$ and $notAdj(a, c, QEG\_POP)$ **do**
  $W \leftarrow 0$;
  **for** $i = 1$ to $N$ **do**
    *//one v-structure found in $EG_i$*
    **if** $a \rightarrow b \leftarrow c \in EG_i$ **then**
      $W \leftarrow W + 1$;
    **end if**
    *//one half edge of v-structure found in $EG_i$*
    **if** $a \rightarrow b \in EG_i$ **then**
      $W \leftarrow W + \frac{1}{2}$;
    **end if**
    *//another half edge of v-structure found in $EG_i$*
    **if** $b \leftarrow c \in EG_i$ **then**
      $W \leftarrow W + \frac{1}{2}$;
    **end if**
  **end for**
  *//calculate and normalize the total weight of v-structure*
  $W \leftarrow W / [N * Max(w_1(u.edges(a, b)), w_1(u.edges(c, b)))]$;
  *//filter noisy-v-structure from QEG_POP*
  **if** $W > \epsilon$ **then**
    *//direct v-structure in QEG_POP*
    orient $a \rightarrow b \leftarrow c \in QEG\_POP$;
    *//set weight for v-structure in QEG_POP*
    $w_2(a, b, c, QEG\_POP) \leftarrow W$;
  **end if**
**end for**
*// add inferred edges*
**for** Each inferred edge found in QEG_POP **do**
  Apply Meek rules;
**end for**
**return** Quality of rndDAG_POP $\leftarrow$ EVAL(QEG_POP)

### D. Extended version of edit distance based evaluation methods for QEG

In order to evaluation the quality of the population, we must calculate now the distance between the QEG in comparison to the EG of the golden graph. This distance is based on the version of distance SHD presented in $II.B.4$ and we add a more *weighted cost*, $\xi$, to each edit cost for transforming QEG to EG of the golden graph. $\xi_{ij}(E(X_i, X_j)) = 1 - w(E(X_i, X_j))$, where $w(E(X_i, x_j))$ is the weight of edge $E(X_i, x_j)$ of QEG. Figure 2 shows these edit costs for the transformations between different kinds of edges:

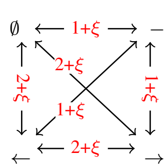
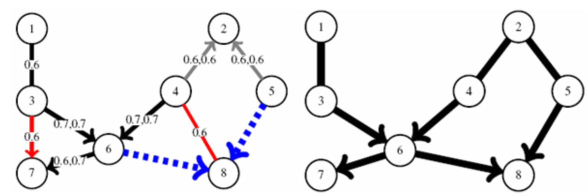



**FIGURE 2:** The edit costs of SHD between QEG and EG of golden graph: $\emptyset$: without edge; $-$: undirected edge; $\rightarrow$: edge oriented right; $\leftarrow$: edge oriented left; $\emptyset \Leftrightarrow -$: *cost* $=1+\xi$; $- \Leftrightarrow \leftarrow$: *cost* $=1+\xi$; $- \Leftrightarrow \rightarrow$: *cost* $=1+\xi$; $\emptyset \Leftrightarrow \rightarrow$: *cost* $=2+\xi$; $\emptyset \Leftrightarrow \leftarrow$: *cost* $=2+\xi$; $\rightarrow \Leftrightarrow \leftarrow$: *cost* $=2+\xi$;

### E. Visualization for QEG based method

The visualization for QEG based method bases on the principle of the graph comparaison method. Originally, the graph comparison is a method for results representation of the *graph matching*. It use the color to spot the graphical differences between one graph in comparison to the other. In the context of BN structure learning, the graph comparaison method is used to illustrate the differences between learned BN and the golden BN. In general, the idea of this method is simple and easy to implement. It is an important to use different colors for different kind of edges: need to add, need to delete, need to reverse and verified (matched). A major constraint to check is that the learned BN and the golden BN should have the same set of nodes and the placement of nodes should have the same coordination. The following figure illustrates this method:



(a) Learned graph (b) Golden graph

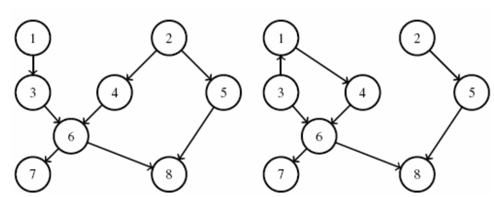
**FIGURE 3:** Comparison of learned colored graph and the golden graph.
**Interpretation:** Line types and colors represent the state of the edges/nodes: "*continuous-line*": edges/nodes verified golden graph, "*dashed-line*": edges/nodes need to add to learned graph; **Black** = edges/nodes verified; Blue = edges/nodes need to add; Red = edges/nodes need to delete; Gray = edges need to reverse.

**Decision:** The more **black** and the less **colored** edges/nodes, the BN is better.

## III. Experimentation and results

*1) Context and protocol of experimentation:* In order to generate a population of networks, we applied the genetic algorithm for BN structure learning by using two C++ libraries: `EO` (Evolving Object), `ProBT_SLP` ($ProBT$ Structure Learning Package) and `Graphiz` for some available methods: *genetic algorithm, scoring, KL divergence* and *visualization*. We also implemented the rest of methods, including: *sensibility/specificity*, *edit distance* and *QEG-EVAL*.

To evaluate the performance of each evaluation method, we used two golden networks: $ASIA$ and $ASIA*$ (add, delete, inverse some edges of $ASIA$) to generate some datasets with different sizes as learning data.



(a) ASIA golden network (b) ASIA* golden network

**FIGURE 4:** Two golden networks used to generate different learning datasets.

From this, we compare the results of each method by learning with different datasets. It allows to observe the behaviour of each evaluation method. Especially, for $QEG - EVAL$, we learned from *Saccharomyces cerevisiae* gene expression data (GDS2350 at GEO (Gene Expression Omnibus) server) with 11 selected genes and 50 test samples. We used a golden structure presented in [19] (see Figure $10.a$). The obtained result allows to verify the quality of the structure learning algorithm in comparison to the others works in the literature.

*2) Results:* The Figure $5-9$ present the results identified by means of evaluation with *statistical approach*. By comparing the change of the different value of each measures, the results show that:

- the score based method can recognize perfectly the change of sample size.
- the sensitivity and specificity based methods can also recognize the effect of change sample size. They are easy to understand how different or similar between learned network and golden network.
- the method based on Kullback-Leibler divergence seems the worst method. Because it clearly shows the uncertainty of its result. For example, the value of KL very ideal in theory, $0.03 - 0.04$, but we can not decide that the learned BN verify "perfectly" the golden BN.
- the method based on edit distance shows its facility and its performance in term of methodology. We can differentiate easily and precisely each structural dissimilarity between two networks.

In general, all methods based on statistical approach allow to obtain only means evaluations. They do not give enough information about the structural quality of population. Almost methods base on the golden network given by experts. The evaluation method based on the comparison to this network needs to take into account the problem of equivalence class (see TABLE 1 for a global view of methods).

Figure $10.b$ presents the result of QEG based evaluation method. With a set of properties such as, line types, edge colors, edge weight/size, arrow head weight/size, the visualization result offer a rich information about the population. This allows to more easily situate the similarity/difference between structures. This also helps the non-Bayesian experts to infer statistically not only each relationship between vertices via the edges intensity, but also precisely the reliability of edges orientation by a set of fondamental Meek rules.

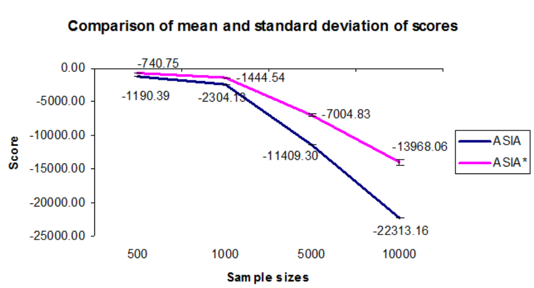



**FIGURE 5**: Results of Score based method.

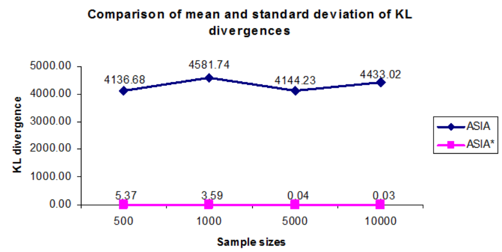



**FIGURE 6**: Results of KL divergence based method.

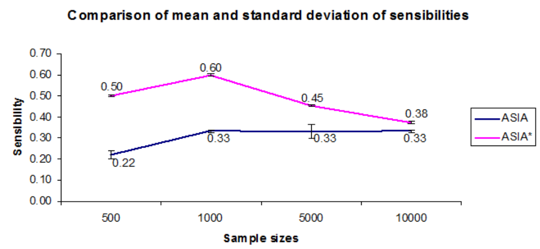



**FIGURE 7**: Results of Sensibility based method.

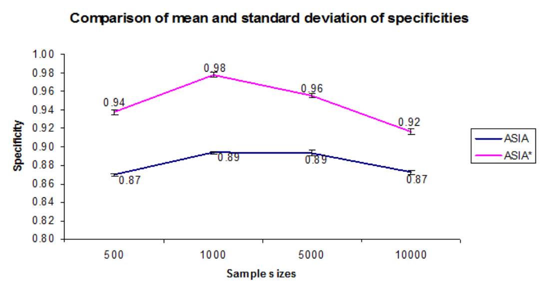



**FIGURE 8**: Results of Specificity based method.

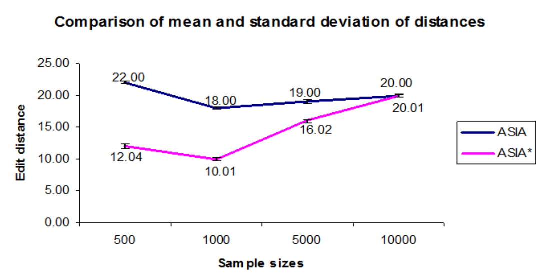



**FIGURE 9**: Results of Edit distance based method.

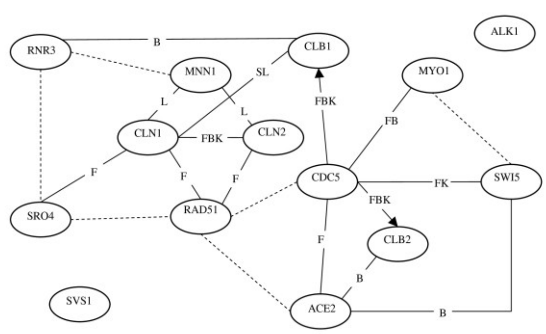



**FIGURE 10.a** Golden structure presented by Ko et al. 2009 [19].

*Legend:* Solid lines and arrows denote direct relationships predicted and confirmed in the BioGRID (B), SGD (S), KEGG (K) databases, literature (L) or by Friedman et al. (F) [20]. Dashed lines denote indirect relationships predicted and confirmed in the databases. Arrows denote directional relationships reported in the KEGG pathway. Lines denote non-directional relationships reported in the KEGG pathway.

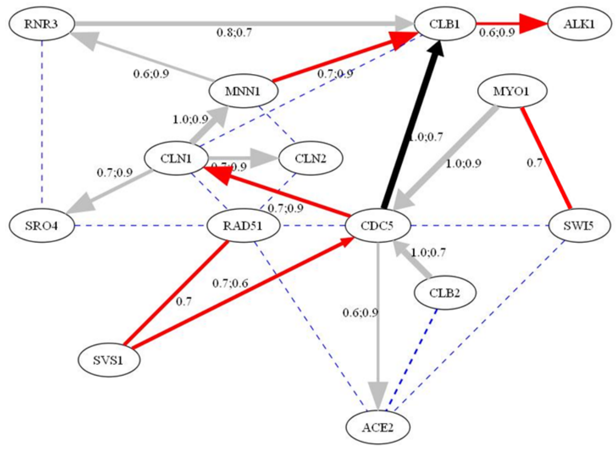



**FIGURE 10.b**: QEG of population of BN learned from Ko golden graph (Figure 7.a): *Legend:* Line types and colors represent the state of the edges/nodes: "*continuous-line*": edges/nodes verified golden graph, "*dashed-line*": edges/nodes need to add to learned graph; **Black** = edges/nodes verified; Blue = edges/nodes need to add; Red = edges/nodes need to delete; Gray = edges need to reverse. The weight of each edge contains two parts: first part is the occurrence frequency of undirected part, second part is the occurrence frequency of arrow head that is associated to the mean of weights of v-structures that contain it. The size of edge and arrow head represent their weights.

*Decision:* By application of extended version of edit distance for QEG,

$$SHD(QEG, EG(Ko)) = 21.5$$

| Methods | Equiv. class | Complex. | Indep. to GS. | Indep. to data |
|---|---|---|---|---|
| 1. Score | $-$ | $-$ | $+$ | $-$ |
| 2. KL divergence | $+$ | $-$ | $-$ | $+$ |
| 3. Sensib./Specif. | $-$ | $+$ | $-$ | $+$ |
| 4. Edit distance | $-$ | $+$ | $-$ | $+$ |

**TABLE 1**: Global view of methods
*(Equiv. class)*: if the method is a solution for the problem of equivalence class; *(Complex.)*: if the calculation of method is complex; *(Indep. to GS.)*: if it the method depends on the golden structure; *(Indep. to data)*: if it the method depends on data; $(+)$: favorable; $(-)$: unfavorable

## IV. Conclusion

The paper describes two evaluation approaches for evaluation of a population of Bayesian networks. We present the details of the algorithms and their implementation and explain how to resolve effectively some crucial problems in evaluation for a population of structures such as Markov-equivalent class, edge orientation. The experimental results identified the helpfulness of proposed methods. They has been presented in the context of the evaluation of quality of a population of Bayesian networks. But they can be also applied to the evaluation of another kind of structures based on directed acyclic graph. We implemented and presented an extended version of edit distance for the epsilon-quasi essential graph based method. Perspectively, the extension of others well-known evaluation methods as score, Kullback Leibler divergence and sensibility/specificity will be also developed especially for epsilon-quasi essential graph.

## Acknowledgment

This research has been supported by the grants BIL (Bio-Informatics Ligérienne, Region of Pays-de-la-Loire, France).

## References

[1] P. Larranaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers, "Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 912–926, 1996.
[2] N. Friedman, M. Goldszmidt, and A. Wyner, "Data analysis with bayesian networks: A bootstrap approach," *Proc. Fifteenth Conf. on Uncertainty in Artificial Intelligence (UAI)*, 1999.
[3] W. Lam and F. Bacchus, "Learning bayesian belief networks. an approach based on the mdl principle," *Computational Intelligence*, vol. 10, p. 269293, 1994.
[4] L. de Campos, "A scoring function for learning bayesian networks based on mutual information and conditional independence tests," *J. Mach. Learn. Res.*, vol. 7, pp. 2149–2187, 2006.
[5] D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks," *Bioinformatics*, 2003.
[6] I. Tsamardinos, L. Brown, and C. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Machine Learning*, vol. 65, no. 1, pp. 31–78, 2006.
[7] S. Imoto, S. Y. Kim, H. Shimodaira, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano, "Bootstrap Analysis of Gene Networks Based on Bayesian Networks and Nonparametric Regression," *Genome Informatics*, vol. 13, pp. 369–370, 2002.
[8] T. Kamimura, H. Shimodaira, S. Imoto, S. Kim, K. Tashiro, S. Kuhara, and S. Miyano, "Multiscale bootstrap analysis of gene networks based," *Genome Informatics*, vol. 14, pp. 350–351, 2003.
[9] C. Rodriguez, "The abc of model selection: Aic, bic and the new cic," *AIP Conference Proceedings*, vol. 803, no. 1, pp. 80–87, 2005.
[10] Bouckaert, R. R., "Probabilistic Network Construction Using the Minimum Description Length Principle," *Lecture Notes in Computer Science*, vol. 747, pp. 41–48, 1993.
[11] D. Heckerman and D. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, pp. 20–197, 1995.
[12] A. M. Carvalho, "Scoring functions for learning bayesian networks," *Inesc-id Tec. Rep.*, 2009.
[13] D. Chickering, "Learning equivalence classes of bayesian-network structure," *Journal of Machine Learning Research*, vol. 2, pp. 445–498, 2002.
[14] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22(1), pp. 79–86, 1951.
[15] H. Bunke, "On a relation between graph edit distance and maximum common subgraph," *Pattern Recogn. Lett.*, vol. 18, no. 9, pp. 689–694, 1997.
[16] T. Verma and J. Pearl, "Equivalence and synthesis of causal models," *Proc. of the 6th Conf. on Uncert. in Art. Intel.*, pp. 220–227, 1990.
[17] M. D. Andersson, S. A. and M. D. Perlman, "A characterization of markov equivalence classes for acyclic digraphs," *Annals of Statistics*, vol. 25, pp. 505–541, 1997.
[18] C. Meek, "Causal inference and causal explanation with background knowledge," *Proc of the 11th Conf on Uncert. in Art. Intel.*, pp. 403–410, 1995.
[19] Y. Ko, C. Zhai, and S. Rodriguez-Zas, "Inference of gene pathways using mixture bayesian networks," *BMC Systems Biology*, vol. 3, no. 1, p. 54, 2009.
[20] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Comp. Molecular Bio.*, pp. 127–135, 2000.